Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	4	clients same servers same queue same accumulate$2	USPAT	OR	OFF	2005/03/15 17:09
L2	0	sychronizate$4 adj protect	USPAT	OR	OFF	2005/03/15 17:09
L3	0	sychronization adj protect	USPAT	OR	OFF	2005/03/15 17:09
L4	1	sychronization adj protect$4	USPAT	OR	OFF	2005/03/15 17:09
L5	1	sychronization adj protect$4	USPAT	OR	ON	2005/03/15 17:09
L6	76	queue same (load balance$3) same accumulate$2	USPAT	OR	ON	2005/03/15 17:10
L7	0	queue same (load balance$3) same accumulate$2 same sychronization	USPAT	OR	ON	2005/03/15 17:10
L8	76	queue same (load balance$3) same accumulate$2 andsychronization	USPAT	OR	ON	2005/03/15 17:10
L9	0	queue same (load balance$3) same accumulate$2 and sychronization	USPAT	OR	ON	2005/03/15 17:10
S1	3657	clients same servers	USPAT	OR	OFF	2005/03/15 17:08
S2	4	clients same servers same queue same accumulate$3	USPAT	OR	OFF	2005/03/15 17:08
S3	12	clients same servers same queue same (load balance$3)	USPAT	OR	OFF	2003/04/04 15:39
S4	0	clients same servers same queue same (load balance$3) same accumulate$3	USPAT	OR	OFF	2003/04/04 15:39
S5	56	queue same (load balance$3) same accumulate$3	USPAT	OR	OFF	2005/03/15 17:10
S6	0	clients same queue same (load balance$3) same accumulate$3	USPAT	OR	OFF	2003/04/04 15:39
S7	1	server same queue same (load balance$3) same accumulate$3	USPAT	OR	OFF	2003/04/04 15:41
S8	0	clients same server same queue same (load balance$3) same fifo	USPAT	OR	OFF	2003/04/04 15:43
S9	149	clients same server same queue same (fifo or first in first out)	USPAT	OR	OFF	2003/04/04 15:50
S10	45	"6012083"	USPAT	OR	OFF	2003/04/07 13:57
S11	1	("5978773").PN.	USPAT; USOCR	OR	OFF	2003/04/07 13:57
S12	1	("5355146").PN.	USPAT; USOCR	OR	OFF	2003/04/09 10:29
S13	6	clients same worker adj threads	USPAT	OR	OFF	2004/01/02 16:10
S14	0	clients same (worker adj threads) same queues	USPAT	OR	OFF	2004/01/02 15:51
S15	0	queue adj in adj queues	USPAT	OR	OFF	2004/01/02 15:51
S16	11302	queues	USPAT	OR	OFF	2004/01/02 15:52

S17	6	clients same (worker adj threads)	USPAT	OR	OFF	2004/01/02 16:09
S18	1	worker adj threads same connect$4 with host	USPAT	OR	ON	2004/01/02 16:12
S19	9	threads adj pool same connect$4 same host	USPAT	OR	ON	2004/01/02 16:17
S20	0	calculat$4 near threads adj connect$4 adj host	USPAT	OR	ON	2004/01/02 16:18
S21	0	calculat$4 near threads adj connect$4	USPAT	OR	ON	2004/01/02 16:18
S22	95	calculat$4 near threads	USPAT	OR	ON	2004/01/02 16:18
S23	45	calculat$4 adj threads	USPAT	OR	ON	2004/01/02 16:20
S24	0	calculat$4 near (threads adj connect$4)	USPAT	OR	ON	2004/01/02 16:18
S25	0	calculat$4 near threads near connect$4	USPAT	OR	ON	2004/01/02 16:19
S26	2	calculat$4 near threads with connect$4	USPAT	OR	ON	2004/01/02 16:19
S27	0	(calculat$4 adj threads) and connect$4 adj host	USPAT	OR	ON	2004/01/02 16:20
S28	40	calculat$4 adj threads and connect$4	USPAT	OR	ON	2004/01/02 16:20
S29	1	calculat$4 adj threads same connect$4	USPAT	OR	ON	2004/01/02 16:21
S30	1	(calculat$4 adj threads) same connect$4	USPAT	OR	ON	2004/01/02 16:22
S31	8	(calculat$4 adj threads) and connect$4 same host	USPAT	OR	ON	2004/01/02 16:24
S32	12	threads adj connect$4 same host	USPAT	OR	ON	2004/01/02 16:36
S33	35	wide adj queues	USPAT	OR	ON	2004/01/02 16:36
S34	35	wide adj queue	USPAT	OR	ON	2004/01/02 16:36
S35	35	wide adj queue and queues	USPAT	OR	ON	2004/01/02 16:36
S36	35	wide adj queue with queues	USPAT	OR	ON	2004/01/02 16:38
S37	0	wide adj queue with queues same (worker adj threads)	USPAT	OR	ON	2004/01/02 16:36
S38	2	wide adj queue with queues same threads	USPAT	OR	ON	2004/01/02 16:37
S39	0	wide adj queue with queues same host and threads	USPAT	OR	ON	2004/01/02 16:37
S40	3	wide adj queue with queues same host	USPAT	OR	ON	2004/01/02 16:37
S41	35	wide adj queue and queues	USPAT	OR	ON	2004/01/02 16:38
S42	0	wide adj queue and queues and host and threads	USPAT	OR	ON	2004/01/02 16:39

S43	0	wide adj queue and queues and host and thread	USPAT	OR	ON	2004/01/02 16:39
S44	15	wide adj queue and queues and host	USPAT	OR	ON	2004/01/02 16:39
S45	1	("6285656").PN.	USPAT; USOCR	OR	OFF	2004/01/06 09:30
S46	1	("5136498").PN.	USPAT; USOCR	OR	OFF	2004/01/06 09:30
S47	17469	clients same servers	USPAT	OR	ON	2004/01/06 16:25
S48	4	$10wanit.xp. and thread	USPAT	OR	ON	2004/01/06 16:25
S49	1146	thread same queue	USPAT	OR	ON	2004/01/06 16:26
S50	830	thread with queue	USPAT	OR	ON	2004/01/06 16:26
S51	23	(thread with queue) same (plurality near2 queue)	USPAT	OR	ON	2004/01/06 16:26
S52	23	((thread with queue) same (plurality near2 queue)) and (@ad<"20000523" @rlad<"20000523")	USPAT	OR	ON	2004/01/06 16:46
S53	0	(((thread with queue) same (plurality near2 queue)) and (@ad<"20000523" @rlad<"20000523")) and determin$4 adj connect$4	USPAT	OR	ON	2004/01/06 16:50
S54	0	(((thread with queue) same (plurality near2 queue)) and (@ad<"20000523" @rlad<"20000523")) and (connect$4 adj tim$4)	USPAT	OR	ON	2004/01/06 16:51
S55	18324	connect$4 adj tim$4	USPAT	OR	ON	2004/01/06 16:52
S56	71	connect$4 adj tim$4 same thread	USPAT	OR	ON	2004/01/06 16:52
S57	0	thread near connect$4 adj tim$4	USPAT	OR	ON	2004/01/06 16:52
S58	71	thread same connect$4 adj tim$4	USPAT	OR	ON	2004/01/06 16:52
S59	24	thread with connect$4 adj tim$4	USPAT	OR	ON	2004/01/06 16:58
S60	0	thread with determing adj time adj connect$4	USPAT	OR	ON	2004/01/06 16:58
S61	0	thread with (determing adj time adj connect$4)	USPAT	OR	ON	2004/01/06 16:58
S62	0	thread with (determing adj connect$4)	USPAT	OR	ON	2004/01/06 16:58
S63	12	thread with (determin$3 adj connect$4)	USPAT	OR	ON	2004/01/06 17:04
S64	1	"5179702".PN.	USPAT	OR	OFF	2004/01/06 16:59
S65	1	("6401010").PN.	USPAT; USOCR	OR	OFF	2004/01/06 17:01
S66	1	"4117459".PN.	USPAT	OR	OFF	2004/01/06 17:01

S67	1	"4282575".PN.	USPAT	OR	OFF	2004/01/06 17:01
S68	1	"4834231".PN.	USPAT	OR	OFF	2004/01/06 17:01
S69	1	"5822216".PN.	USPAT	OR	OFF	2004/01/06 17:02
S70	1	"6324520".PN.	USPAT	OR	OFF	2004/01/06 17:02
S71	0	determin$4 adj thread adj connect$4	USPAT	OR	ON	2004/01/06 17:04
S72	4861	thread adj connect$4	USPAT	OR	ON	2004/01/06 17:04
S73	32	determin$4 with (thread adj connect$4)	USPAT	OR	ON	2004/01/06 17:05
S74	1	determin$4 with (thread adj connect$4) with tim$4	USPAT	OR	ON	2004/01/06 17:05
S75	105	(thread adj connect$4) with tim$4	USPAT	OR	ON	2004/01/06 17:06
S76	0	thread adj connect$4 adj tim$4	USPAT	OR	ON	2004/01/06 17:06
S77	105	(thread adj connect$4) with tim$4	USPAT	OR	ON	2004/01/06 17:06
S78	17	(thread adj connect$4) near3 tim$4	USPAT	OR	ON	2004/01/06 17:09
S79	13	(thread adj (reconnect or session))	USPAT	OR	ON	2004/01/07 08:06
S80	13	(thread adj (reconnect or session or logon))	USPAT	OR	ON	2004/01/07 08:09
S81	13	(thread adj (reconnect or session or relogon))	USPAT	OR	ON	2004/01/07 16:20
S82	4	$10wanit.xp. and thread	USPAT	OR	ON	2004/01/07 09:07
S83	0	$10wanit.xp. and thread and (time adj stamp)	USPAT	OR	ON	2004/01/07 09:07
S84	0	thread same queue same (tim$4 adj stamp) same plurality near3 queue	USPAT	OR	ON	2004/01/07 09:08
S85	0	thread same queue same (tim$4 adj stamp) same (plurality near3 queue)	USPAT	OR	ON	2004/01/07 09:08
S86	9	thread same queue same (tim$4 adj stamp)	USPAT	OR	ON	2004/01/07 09:17
S87	40	thread with (tim$4 adj stamp)	USPAT	OR	ON	2004/01/07 09:17
S88	24	thread with (tim$4 adj stamp) and queues	USPAT	OR	ON	2004/01/07 09:17
S89	5	thread with (tim$4 adj stamp) and plurality same queues	USPAT	OR	ON	2004/01/07 09:17
S90	13	(thread adj (reconnect or session or relogon))	USPAT	OR	ON	2004/01/07 16:20
S91	13	(thread adj (reconnect or session or logon))	USPAT	OR	ON	2004/01/07 16:21
S92	3	(thread adj reconnect$3)	USPAT	OR	ON	2004/01/07 16:22
S93	0	(thread adj reconnect$3) same log adj on	USPAT	OR	ON	2004/01/07 16:23

S94	0	(thread adj rconnect$3) with (count$4 adj time)	USPAT	OR	ON	2004/01/07 16:23
S95	0	(thread adj rconnect$3) same (count$4 adj time)	USPAT	OR	ON	2004/01/07 16:23
S96	0	(thread adj rconnect$3) and (count$4 adj time)	USPAT	OR	ON	2004/01/07 16:24
S97	0	thread adj (count$4 adj time)	USPAT	OR	ON	2004/01/07 16:24
S98	29	thread same (count$4 adj time)	USPAT	OR	ON	2004/01/07 16:24
S99	2	thread same (count$4 adj time) same connect$3	USPAT	OR	ON	2004/01/07 16:25
S100	2218	connect$3 same (count$4 adj time)	USPAT	OR	ON	2004/01/07 16:25
S101	1	connect$3 same (count$4 adj time) with thread	USPAT	OR	ON	2004/01/07 16:26
S102	0	determin$4 adj (count$4 adj time) with thread	USPAT	OR	ON	2004/01/07 16:26
S103	117	determin$4 adj (count$4 adj time)	USPAT	OR	ON	2004/01/07 16:26
S104	0	determin$4 adj (count$4 adj time) same thread	USPAT	OR	ON	2004/01/07 16:26
S105	1	determin$4 adj (count$4 adj time)and thread	USPAT	OR	ON	2004/01/07 16:27
S106	394	determin$4 adj count$4 and thread	USPAT	OR	ON	2004/01/07 16:27
S107	38	determin$4 adj count$4 same thread	USPAT	OR	ON	2004/01/07 16:27
S108	7	determin$4 adj count$4 same thread same connect$3	USPAT	OR	ON	2004/01/07 16:29
S109	70	count$4 same (thread adj connect$3)	USPAT	OR	ON	2004/01/07 16:30
S110	30	count$4 with (thread adj connect$3)	USPAT	OR	ON	2004/01/07 16:30
S111	30	count$4 with (thread adj connect$3)	USPAT	OR	ON	2004/01/07 16:30
S112	5	count$4 near (thread adj connect$3)	USPAT	OR	ON	2004/01/07 16:31
S113	0	count$4 with (thread adj connect$3) same client	USPAT	OR	ON	2004/01/07 16:31
S114	30	count$4 with (thread adj connect$3)	USPAT	OR	ON	2004/01/07 16:31
S115	11	count$4 with (thread adj connect$3) same device	USPAT	OR	ON	2004/01/07 16:34
S116	93	time$4 with (thread adj connect$3)	USPAT	OR	ON	2004/01/07 16:34

S117	0	time$4 with (thread adj connect$3)same host	USPAT	OR	ON	2004/01/07 16:35
S118	0	time$4 with (thread adj connect$3) same host	USPAT	OR	ON	2004/01/07 16:35
S119	0	count$3 same (thread adj reconnect$3)	USPAT	OR	ON	2004/01/07 16:35
S120	3	(thread adj reconnect$3)	USPAT	OR	ON	2004/01/07 16:35
S121	1	("6182109").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:20
S122	1	("5761507").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:20
S123	1	("6427161").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:21
S124	1	("6012083").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:21
S125	1	("6185601").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:21
S126	1	("6145001").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:22
S127	1	("5708834").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:22
S128	1	("5781550").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:22
S129	1	("6078960").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:23
S130	1	("6442685").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:23
S131	1	("5884028").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:23
S132	1	("6170013").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:23
S133	1	("5815662").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:24
S134	1	("6539429").PN.	USPAT; USOCR	OR	OFF	2004/01/08 16:24

Welcome
United States Patent and Trademark Office

Welcome to IEEE Xplore®
- ○ Home
- ○ What Can I Access?
- ○ Log-out

Tables of Contents
- ○ Journals & Magazines
- ○ Conference Proceedings
- ○ Standards

Search
- ○ By Author
- ○ Basic
- ○ Advanced
- ○ CrossRef

Member Services
- ○ Join IEEE
- ○ Establish IEEE Web Account
- ○ Access the IEEE Member Digital Library

IEEE Enterprise
- ○ Access the IEEE Enterprise File Cabinet

Your search matched **1** of **1138071** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enterin new one in the text box.

queue<and>accumulate<and>synchronization Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **Peak rate regulation scheme for ATM networks and its performance**
Ohta, C.; Toda, H.; Yamamoto, M.; Okada, H.; Tezuka, Y.;
INFOCOM '93. Proceedings.Twelfth Annual Joint Conference of the IEEE Compu and Communications Societies. Networking: Foundation for the Future. IEEE , March-1 April 1993
Pages:680 - 689 vol.2

[Abstract] [PDF Full-Text (620 KB)] **IEEE CNF**

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Welcome to IEEE Xplore®

○- Home
○- What Can I Access?
○- Log-out

Tables of Contents

○- Journals & Magazines
○- Conference Proceedings
○- Standards

Search

○- By Author
○- Basic
○- Advanced
○- CrossRef

Member Services

○- Join IEEE
○- Establish IEEE Web Account
○- Access the IEEE Member Digital Library

IEEE Enterprise

○- Access the IEEE Enterprise File Cabinet

Your search matched **2** of **1138071** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enterin new one in the text box.

queue<and>synchronization<and>protect Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **Parallel Dispatch Queue: a queue-based programming abstraction to parallelize fine-grain communication protocols**
Falsafi, B.; Wood, D.A.;
High-Performance Computer Architecture, 1999. Proceedings. Fifth Internation Symposium On , 9-13 Jan. 1999
Pages:182 - 192

[Abstract] [PDF Full-Text (112 KB)] **IEEE CNF**

2 **Hardware support for release consistency with queue-based synchronization**
Jae Bum Lee; Chu Shik Jhon;
Parallel and Distributed Systems, 1997. Proceedings., 1997 International Conference on , 10-13 Dec. 1997
Pages:144 - 151

[Abstract] [PDF Full-Text (696 KB)] **IEEE CNF**

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

IEEE HOME | SEARCH IEEE | SHOP | WEB ACCOUNT | CONTACT IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.8

Welcome
United States Patent and Trademark Office

Help FAQ Terms IEEE Peer Review **Quick Links**

» Sea

Welcome to IEEE *Xplore*®

- ○ Home
- ○ What Can I Access?
- ○ Log-out

Tables of Contents

- ○ Journals & Magazines
- ○ Conference Proceedings
- ○ Standards

Search

- ○ By Author
- ○ Basic
- ○ Advanced
- ○ CrossRef

Member Services

- ○ Join IEEE
- ○ Establish IEEE Web Account
- ○ Access the IEEE Member Digital Library

IEEE Enterprise

- ○ Access the IEEE Enterprise File Cabinet

Your search matched **0** of **1138071** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enterin new one in the text box.

queues<and>sychronization **Search**

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:
No documents matched your query.

Print Format

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ | Terms | Back to Top

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

queues and sychronization and protect and threads and accum SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
queues and **sychronization** and **protect** and **threads** and **accumulate**

Found **8,611** of **151,219**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Implementing Ada protected objects—interface issues and optimization
E. W. Giering, T. P. Baker
November 1995 **Proceedings of the conference on TRI-Ada '95: Ada's role in global markets: solutions for a changing complex world**
Full text available: pdf(1.13 MB) Additional Information: full citation, references, citings

2 Multi-model parallel programming in psyche
M. L. Scott, T. J. LeBlanc, B. D. Marsh
February 1990 **ACM SIGPLAN Notices , Proceedings of the second ACM SIGPLAN symposium on Principles & practice of parallel programming**, Volume 25 Issue 3
Full text available: pdf(1.48 MB) Additional Information: full citation, abstract, references, citings, index terms

Many different parallel programming models, including lightweight processes that communicate with shared memory and heavyweight processes that communicate with messages, have been used to implement parallel applications. Unfortunately, operating systems and languages designed for parallel programming typically support only one model. Multi-model parallel programming is the simultaneous use of several different models, both across programs and within a single program. This paper describes mu ...

3 A resource management framework for priority-based physical-memory allocation
Kingsley Cheung, Gernot Heiser
January 2002 **Australian Computer Science Communications , Proceedings of the seventh Asia-Pacific conference on Computer systems architecture - Volume 6**, Volume 24 Issue 3
Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, references, index terms

Most multitasking operating systems support scheduling priorities in order to ensure that processor time is allocated to important or time-critical processes in preference to less important ones. Ideally this would prevent a low-priority process from slowing the execution of a high-priority one. In practice, strict prioritisation is undermined by a lack of suitable allocation policy for resources other than CPU time. For example, a low priority process may degrade the execution speed of a high-p ...

4 Guide for the use of the Ada Ravenscar Profile in high integrity systems

Alan Burns, Brian Dobbing, Tullio Vardanega
June 2004 **ACM SIGAda Ada Letters**, Volume XXIV Issue 2
Full text available: pdf(548.17 KB) Additional Information: full citation, references

5 Using threads in interactive systems: a case study
Carl Hauser, Christian Jacobi, Marvin Theimer, Brent Welch, Mark Weiser
December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles**, Volume 27 Issue 5
Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references, citings, index terms

We describe the results of examining two large research and commercial systems for the ways that they use threads. We used three methods: analysis of macroscopic thread statistics, analysis the microsecond spacing between thread events, and reading the implementation code. We identify ten different paradigms of thread usage: *defer work, general pumps, slack processes, sleepers, one-shots, deadlock avoidance, rejuvenation, serializers, encapsulated fork and exploiting parallelism*. While so ...

6 The rendezvous is dead—long live the protected object
Dragan Macos, Frank Mueller
November 1998 **ACM SIGAda Ada Letters , Proceedings of the 1998 annual ACM SIGAda international conference on Ada**, Volume XVIII Issue 6
Full text available: pdf(750.79 KB) Additional Information: full citation, references, index terms

7 Design challenges of virtual networks: fast, general-purpose communication
Alan M. Mainwaring, David E. Culler
May 1999 **ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 34 Issue 8
Full text available: pdf(1.57 MB) Additional Information: full citation, abstract, references, citings, index terms

Virtual networks provide applications with the illusion of having their own dedicated, high-performance networks, although network interfaces posses limited, shared resources. We present the design of a large-scale virtual network system and examine the integration of communication programming interface, system resource management, and network interface operation. Our implementation on a cluster of 100 workstations quantifies the impact of virtualization on small message latencies and throughput ...

Keywords: application programming interfaces, direct network access, high-performance clusters, protocol architecture and implementation, system resource management, virtual networks

8 Remote queues: exposing message queues for optimization and atomicity
Eric A. Brewer, Frederic T. Chong, Lok T. Liu, Shamik D. Sharma, John D. Kubiatowicz
July 1995 **Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures**
Full text available: pdf(1.78 MB) Additional Information: full citation, references, citings, index terms

9

Tera hardware-software cooperation

Gail Alverson, Preston Briggs, Susan Coatney, Simon Kahan, Richard Korry
November 1997 **Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available: pdf(217.50 KB) Additional Information: full citation, abstract, references, citings

The development of Tera's MTA system was unusual. It respected the need for fast hardware and large shared memory, facilitating execution of the most demanding parallel application programs. But at the same time, it met the need for a clean machine model enabling calculated compiler optimizations and easy programming; and the need for novel architectural features necessary to support fast parallel system software. From its inception, system and application needs have molded the MTA architecture. ...

10 Generative communication in Linda

David Gelernter
January 1985 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 7 Issue 1

Full text available: pdf(2.48 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Generative communication is the basis of a new distributed programming langauge that is intended for systems programming in distributed settings generally and on integrated network computers in particular. It differs from previous interprocess communication models in specifying that messages be added in tuple-structured form to the computation environment, where they exist as named, independent entities until some process chooses to receive them. Generative communication results in a number ...

11 Implementation and performance of Munin

John B. Carter, John K. Bennett, Willy Zwaenepoel
September 1991 **ACM SIGOPS Operating Systems Review , Proceedings of the thirteenth ACM symposium on Operating systems principles**, Volume 25 Issue 5

Full text available: pdf(1.46 MB) Additional Information: full citation, abstract, references, citings, index terms

Munin is a distributed shared memory (DSM) system that allows shared memory parallel programs to be executed efficiently on distributed memory multiprocessors. Munin is unique among existing DSM systems in its use of *multiple consistency protocols* and in its use of *release consistency*. In Munin, shared program variables are annotated with their expected access pattern, and these annotations are then used by the runtime system to choose a consistency protocol best suited to that acc ...

12 Performance measurements for multithreaded programs

Minwen Ji, Edward W. Felten, Kai Li
June 1998 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems**, Volume 26 Issue 1

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

Multithreaded programming is an effective way to exploit concurrency, but it is difficult to debug and tune a highly threaded program. This paper describes a performance tool called Tmon for monitoring, analyzing and tuning the performance of multithreaded programs. The performance tool has two novel features: it uses "thread waiting time" as a measure and constructs thread waiting graphs to show thread dependencies and thus performance bottlenecks, and it identifies "semi-busy-waiting" points w ...

13 Multiprocessor main memory transaction processing

K. Li, J. F. Naughton
January 2000 **Proceedings of the first international symposium on Databases in parallel and distributed systems**
Full text available: pdf(1.16 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we describe an experiment designed to evaluate the potential transaction processing system performance achievable through the combination of multiple processors and massive memories. The experiment consisted of the design and implementation of a transaction processing kernel on stock multiprocessors. We found that with sufficient memory, multiple processors can greatly improve performance. A prototype implementation of the kernel on a pair of Firefly multiprocessors (each with ...

14 Transient-fault recovery for chip multiprocessors

Mohamed Gomaa, Chad Scarbrough, T. N. Vijaykumar, Irith Pomeranz
May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2
Full text available: pdf(370.75 KB) Additional Information: full citation, abstract, references

To address the increasing susceptibility of commodity chip multiprocessors (CMPs) to transient faults, we propose Chiplevel Redundantly Threaded multiprocessor with Recovery (CRTR). CRTR extends the previously-proposed CRT for transient-fault detection in CMPs, and the previously-proposed SRTR for transient-fault recovery in SMT. All these schemes achieve fault tolerance by executing and comparing two copies, called leading and trailing threads, of a given application. Previous recovery schemes ...

15 Speculative synchronization: applying thread-level speculation to explicitly parallel applications

José F. Martínez, Josep Torrellas
October 2002 **Proceedings of the 10th international conference on Architectural support for programming languages and operating systems**, Volume 36 , 30 , 37 Issue 5 , 5 , 10
Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings

Barriers, locks, and flags are synchronizing operations widely used programmers and parallelizing compilers to produce race-free parallel programs. Often times, these operations are placed suboptimally, either because of conservative assumptions about the program, or merely for code simplicity.We propose *Speculative Synchronization,* which applies the philosophy behind Thread-Level Speculation (TLS) to explicitly parallel applications. Speculative threads execute past active barriers, busy ...

16 A survey of processors with explicit multithreading

Theo Ungerer, Borut Robič, Jurij Šilc
March 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 1
Full text available: pdf(920.16 KB) Additional Information: full citation, abstract, references, index terms

Hardware multithreading is becoming a generally applied technique in the next generation of microprocessors. Several multithreaded processors are announced by industry or already into production in the areas of high-performance microprocessors, media, and network processors.A multithreaded processor is able to pursue two or more threads of control in parallel within the processor pipeline. The contexts of two or more threads of control are often stored in separate on-chip register sets. Unused i ...

Keywords: Blocked multithreading, interleaved multithreading, simultaneous multithreading

17 Scheme fair threads

Manuel Serrano, Fréderic Boussinot, Bernard Serpette
August 2004 **Proceedings of the 6th ACM SIGPLAN international conference on Principles and practice of declarative programming**

Full text available: pdf(236.87 KB) Additional Information: full citation, abstract, references, index terms

This paper presents *Fair Threads*, a new model for concurrent programming. This multi-threading model combines preemptive and cooperative scheduling. *User* threads execute according to a cooperative strategy. *Service* threads execute according to a preemptive strategy. User threads may ask services from service threads in order to improve performance by exploiting hardware parallelism and in order to execute non-blocking operations.Fair threads are experimented within the conte ...

Keywords: concurrency, functional languages, scheme, threads

18 Portable resource control in Java

Walter Binder, Jane G. Hulaas, Alex Villazón
October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**, Volume 36 Issue 11

Full text available: pdf(307.08 KB) Additional Information: full citation, abstract, references, citings, index terms

Preventing abusive resource consumption is indispensable for all kinds of systems that execute untrusted mobile coee, such as mobile object sytems, extensible web servers, and web browsers. To implement the required defense mechanisms, some support for resource control must be available: accounting and limiting the usage of physical resources like CPU and memory, and of logical resources like threads. Java is the predominant implementation language for the kind of systems envisaged here, even th ...

Keywords: Java, bytecode rewriting, micro-kernels, mobile object systems, resource control, security

19 Performance counters and state sharing annotations: a unified approach to thread locality

Boris Weissman
October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 33 , 32 Issue 11 , 5

Full text available: pdf(1.76 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a combined approach for improving thread locality that uses the bardware performance monitors of modem processors and program-centric code annotations to guide thread scheduling on SMPs. The approach relies on a shared state cache model to compute expected thread footprints in the cache on-line. The accuracy of the model has been analyzed by simmations involving a set of parallel applications. We demonstrate how the cache model can be used to implement several practical loca ...

20 Techniques to Reduce the Soft Error Rate of a High-Performance Microprocessor

March 2004 **ACM SIGARCH Computer Architecture News , Proceedings of the 31st annual international symposium on Computer architecture**, Volume 32 Issue 2

Full text available: pdf(228.67 KB) Additional Information: full citation, abstract

Transient faults due to neutron and alpha particle strikes posea significant obstacle to increasing processor transistor counts infuture technologies. Although fault rates of individual transistorsmay not rise significantly, incorporating more transistors into adevice makes that device more likely to encounter a fault. Hence,maintaining processor error rates at acceptable levels will requireincreasing design effort.This paper proposes two simple approaches to reduce errorrates and evaluates thei ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next